Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Please amend claims 3, 4, 11, 15, and 19 as follows:

1. (original): A processor address translation apparatus for translating an instruction operand address to a different operand address, the processor address translation apparatus comprising:

a memory with an address input for selecting a data element from a plurality of data elements;

an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution; and

an address translation unit for accessing the memory in a translation pattern, having the operand address as input and, in response to the instruction received in the instruction register, translating the operand address to form the different operand address in accordance with the translation pattern, the different operand address accessing a data element from the memory through the address input.

2. (original): The processor address translation apparatus of claim 1 wherein the address translation unit further comprises:

a plurality of translation parameters and address translation functions supporting a plurality of translation patterns; and

an input to select a translation pattern from the plurality of supported translation patterns.

3. (currently amended): The processor address translation apparatus of claim 2 wherein the translation parameters include k by k s bits and k e bits for a k bit address and address translation functions further comprises combinatorial logic governed by the following equations, where the operand address ~~inputs are~~ input is A0, A1, ..., A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and translated address output are A0', A1', ..., A(k-1)',

$$\begin{pmatrix} A0' \\ A1' \\ \vdots \\ A(k-1)' \end{pmatrix} = \begin{pmatrix} s0 & s1 & \cdots & s(k-1) & e0 \\ sk & s(k+1) & \cdots & s(2k-1) & e1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(k-1)k & s(k-1)k+1 & \cdots & s(k-1)(k+1) & e(k-1) \end{pmatrix} \times \begin{pmatrix} A0 \\ A1 \\ \vdots \\ A(k-1) \\ 1 \end{pmatrix}.$$

4. (currently amended): The processor address translation apparatus of claim 1 wherein the ~~processor~~ instruction is a block load instruction.

5. (original): The processor address translation apparatus of claim 1 disposed within a plurality of instruction operand address paths for a plurality of instructions, the plurality of instructions fetched for simultaneous execution.

6. (original): The processor address translation apparatus of claim 5 wherein the plurality of instructions constitute a very long instruction word (VLIW).

7. (original): A processor register file indexing (RFI) address translation apparatus for translating an RFI sequence of instruction operand addresses to an RFI sequence of different operand addresses, the processor RFI address translation apparatus comprising:

a memory with an address input for selecting a data element from a plurality of data elements;

an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution;

an RFI update unit enabled to generate on the RFI update unit's output a linear sequence of RFI operand addresses in response to a received sequence of RFI translation type instructions;

a multiplexer for selecting between the operand address from the instruction register for a first RFI operation and selecting the RFI update unit's output for subsequent RFI operations; and

an address translation unit for accessing the memory in a translation pattern, receiving a sequence of operand addresses from the multiplexer and, in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses in accordance with the translation pattern, the different operand addresses each accessing a data element from the memory through the address input.

8. (original): The processor RFI address translation apparatus of claim 7 disposed within PEs of an array of PEs.

9. (original): The processor RFI address translation apparatus of claim 7 disposed within a plurality of instruction operand address paths for a plurality of instructions, the plurality of instructions fetched for simultaneous execution.

10. (original): The processor RFI address translation apparatus of claim 9 wherein the plurality of instructions constitute a very long instruction word (VLIW).

11. (currently amended): An address translation memory device for accessing data at translated addresses, the address translation memory device comprising:

a first read address input;

a storage device having data accessible at addressable locations, a second read address input internal to the address translation memory device for selecting data from the storage device during read operations, and a data output port; and

an address translation unit for accessing the storage device in a translation pattern, the address translation unit translating the first read address input in accordance with the translation pattern, to the storage device second read address input for reading data from the storage device at a translated address during a read operation.

12. (original): The address translation memory device of claim 11 further comprises:

a first write address input;

a storage device having data accessible at addressable locations, a second write address input for selecting data in the storage device during write operations, and a data input port; and

an address translation unit, for accessing the storage device in a translation pattern, the address translation unit translating the first write address input in accordance with the translation

pattern, to the storage device second write address input for writing data to the storage device at a translated address during a write operation.

13. (original): The address translation memory device of claim 11 wherein the storage device further comprises location selection logic merged with the address translation unit.

14. (original): The address translation memory device of claim 11 further comprises:

a plurality of translation parameters and address translation functions supporting a plurality of translation patterns; and

an input to select a translation pattern from the plurality of supported translation patterns.

15. (currently amended): The address translation memory device of claim 14 wherein the translation parameters include k by k s bits and k e bits for a k bit address and address translation functions further comprises combinatorial logic governed by the following equations, where the first read address ~~inputs are~~ input is A0, A1, ..., A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and translated address output are A0', A1', ..., A(k-1)',

$$\begin{pmatrix} A0' \\ A1' \\ \vdots \\ A(k-1)' \end{pmatrix} = \begin{pmatrix} s0 & s1 & \cdots & s(k-1) & e0 \\ sk & s(k+1) & \cdots & s(2k-1) & e1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(k-1)k & s(k-1)k+1 & \cdots & s(k-1)(k+1) & e(k-1) \end{pmatrix} \times \begin{pmatrix} A0 \\ A1 \\ \vdots \\ A(k-1) \\ 1 \end{pmatrix}.$$

16. (original): A processor address translation method for translating an instruction operand address to a different operand address, the address translation method comprising:

receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution;

translating the operand address according to a function; and

accessing a data element with the translated address, and repeating the receiving, translating, and accessing steps to access data elements in a pattern according to the function.

17. (original): The processor address translation method of claim 16 wherein the function comprises combinatorial logic for translating the operand address.

18. (original): An address translation method for translating a first address of a first data element in a memory to a second address of a second data element in the memory, the address translation method comprising:

determining a set of {s, e} bits that specify a translation pattern;

loading the set of {s, e} bits into an address translation parameter control register;

enabling an address translation unit for translation;

initiating a read operation to read a first data element at a first address during a read operation;

translating the first address to the second address in accordance with the {s, e} bit specified translation pattern; and

completing the read operation by reading the second data element at the second address.

19. (currently amended): The address translation method of claim 16 wherein the set of {s, e} bits include k by k s bits and k e bits for a k bit address and address translation functions further comprises combinatorial logic governed by the following equations, where the operand address ~~inputs are~~ is A0, A1, ..., A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and translated address output are A0', A1', ..., A(k-1)',

$$\begin{pmatrix} A0' \\ A1' \\ \vdots \\ A(k-1)' \end{pmatrix} = \begin{pmatrix} s0 & s1 & \cdots & s(k-1) & e0 \\ sk & s(k+1) & \cdots & s(2k-1) & e1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(k-1)k & s(k-1)k+1 & \cdots & s(k-1)(k+1) & e(k-1) \end{pmatrix} \times \begin{pmatrix} A0 \\ A1 \\ \vdots \\ A(k-1) \\ 1 \end{pmatrix}.$$